LAB: SQL INJECTION

LAB: SQL INJECTION

This lab is mainly focused on SQL Injection attack, which is a technique that exploits the database of an application. Here you can learn basic things in SQL like, creating databases, creating tables and inserting the records into the table. Finally, we Inject malicious SQL code to exploit the database.

1. INSTRUCTIONS

This lab is performed on SeedLab's Ubuntu Linux version 16.04. It is pre-built virtual machine which has all the necessary tools to perform this attack.

1.User ID: **seed**, Password: **dees**

2.UserID: **root**, Password: **seedubuntu**.

Note: ubuntu does not allow root to login directly from login window. You have to login as a normal user, and then use the command su to login to the root account.

1.1: What is SQL?

SQL stands for Structured Query Language. It is a standard language to communicate with the relational database management systems. It is used to add, update, retrieve delete data from the database. Some common relational database management systems are Oracle, Microsoft SQL Server and MySQL etc.,

1.2: What is SQL Injection?

SQL Injection is a type of attack where user inputs the applications with malicious SQL code and takes the control over the application's database. Attacker may input the code through a front-end form. When the user input (malicious SQL code) passes to backend it may corrupt the databases.

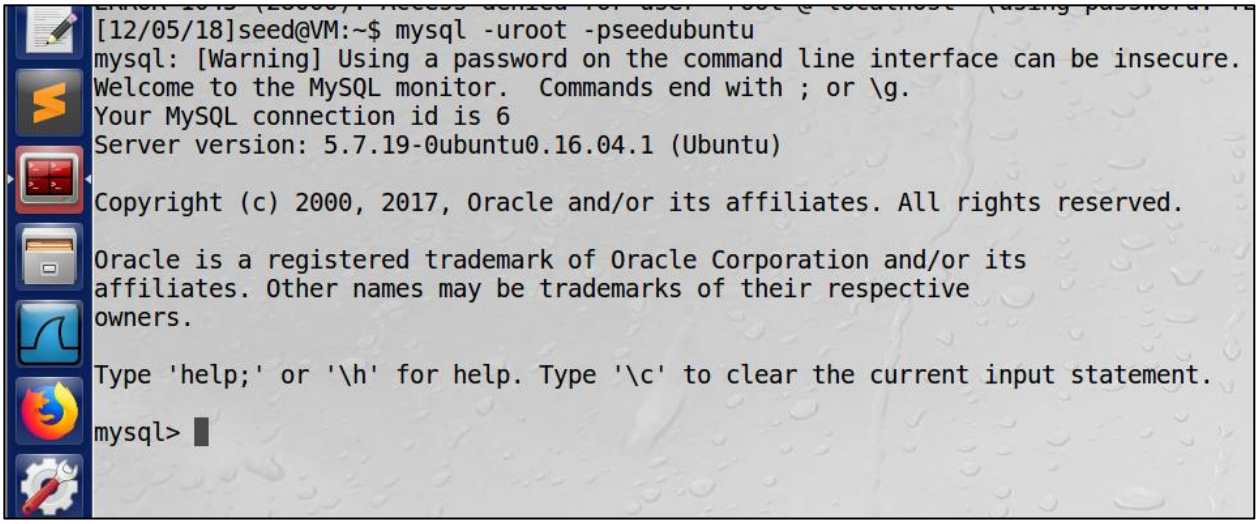
Attacker uses SQL injection to bypass authentication and authorization and retrieve contents of entire database. It is also used to add, modify and delete the records in the database affecting the database integrity.

2. SQL ESSENTIALS

2.1. Log-in to MySQL

Open the 'Terminal' in seedlab's VM and type the following command to login to the MySQL.

```
$mysql -uroot -pseedubuntu
```



2.2. Create a Database

The `show databases;` command is used to list all the existing databases. Then we will create a database called 'student'.

```
mysql> show databases;
```

```
mysql> CREATE DATABASE database_name;
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| Users              |
| elgg_csrf          |
| elgg_xss           |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
+--------------------+
8 rows in set (0.01 sec)

mysql>
```

```
mysql> CREATE DATABASE student;
Query OK, 1 row affected (0.00 sec)

mysql>
```

2.3. Create a Table

We have just created a database called STUDENT. This database is empty at this point. Relational databases store the data using the tables. Let us create a table called 'tbl1'.

```
/bin/bash 168x38
mysql> CREATE TABLE tbl1 ( ID int NOT NULL AUTO_INCREMENT, Name VARCHAR(30) NOT NULL, Department VARCHAR(50) NOT NULL, Password VARCHAR(60), PRIMARY KEY (ID) );
Query OK, 0 rows affected (0.13 sec)
```

```
mysql>CREATE TABLE tbl1 (
      ID int NOT NULL AUTO_INCREMENT,
      Name VARCHAR(30) NOT NULL,
      Department VARCHAR(50) NOT NULL,
      Password VARCHAR(60), PRIMARY KEY(ID));
```

```
mysql> DESCRIBE tbl1;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| ID         | int(11)     | NO   | PRI | NULL    | auto_increment |
| Name       | varchar(30) | NO   |     | NULL    |                |
| Department | varchar(50) | NO   |     | NULL    |                |
| Password   | varchar(60) | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

2.4. Insert Rows

Data is stored in tables in the form of rows. Each row of data is called as a 'Record'. Now let us insert few records into the table. We are not specifying the value for the ID column because we used AUTO_INCREMENT statement for that row, It allows a unique number to be generated when a new record is inserted into a table.

```
mysql> INSERT INTO tbl1 (Name, Department, Password)  VALUES ('Charles', 'Computers', 'pass1');
Query OK, 1 row affected (0.01 sec)
```

We can also insert multiple rows at once.

```
mysql>
mysql> INSERT INTO tbl1 (Name, Department, Password)  VALUES ('Jack', 'Computers', 'pass2'),
    -> ('Tom', 'Mechanical', 'pass3'), ('John', 'Bio-Science', 'pass4'), ('harry', 'Psychology', 'pass5');
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
>INSERT INTO tbl1 (Name, Department, Password)  VALUES
('Charles', 'Computers', 'pass1');
```

```
>INSERT INTO tbl1 (Name, Department, Password)  VALUES ('Jack',
'Computers', 'pass2'),('Tom', 'Mechanical', 'pass3'), ('John',
'Bio-Science', 'pass4'), ('harry', 'Psychology', 'pass5');
```

2.5. Retrieve Data

We use SELECT statements to retrieve data from the database. It is the most simple and common command to retrieve information. In the following example we will retrieve all the records from the database.

```
mysql> SELECT * FROM tbl1;
+----+---------+-------------+----------+
| ID | Name    | Department  | Password |
+----+---------+-------------+----------+
|  1 | Charles | Computers   | pass1    |
|  2 | Jack    | Computers   | pass2    |
|  3 | Tom     | Mechanical  | pass3    |
|  4 | John    | Bio-Science | pass4    |
|  5 | harry   | Psychology  | pass5    |
+----+---------+-------------+----------+
5 rows in set (0.00 sec)
```

2.6. WHERE Clause

WHERE Clause is used filter the records. It sets conditions for several types of SQL statements. Only statements which satisfy the condition are retrieved from the database.

In the following example, the statement retrieves the record whose ID is equal to 5. In our case ID 5 belongs to harry.

```
mysql> SELECT * FROM tbl1 WHERE ID=5;
+----+-------+------------+----------+
| ID | Name  | Department | Password |
+----+-------+------------+----------+
|  5 | harry | Psychology | pass5    |
+----+-------+------------+----------+
1 row in set (0.00 sec)
```

In this example, statement retrieve records whose department is 'Computers'. There were two records that satisfy this condition.

```
mysql> SELECT * FROM tbl1 WHERE Department='Computers';
+----+---------+------------+----------+
| ID | Name    | Department | Password |
+----+---------+------------+----------+
|  1 | Charles | Computers  | pass1    |
|  2 | Jack    | Computers  | pass2    |
+----+---------+------------+----------+
2 rows in set (0.00 sec)
```

If the condition is always True, then it retrieves all the rows.

```
mysql> SELECT * FROM tbl1 WHERE 1=1;
+----+---------+-------------+----------+
| ID | Name    | Department  | Password |
+----+---------+-------------+----------+
|  1 | Charles | Computers   | pass1    |
|  2 | Jack    | Computers   | pass2    |
|  3 | Tom     | Mechanical  | pass3    |
|  4 | John    | Bio-Science | pass4    |
|  5 | harry   | Psychology  | pass5    |
+----+---------+-------------+----------+
5 rows in set (0.01 sec)
```

2.7. UPDATE SQL Statement

Update query is used to modify the existing records in the table. Let us UPDATE the Department of 'Jack'. Right now, it is 'Computers'. Let us change it to 'Electrical'.

```
mysql> UPDATE tbl1 SET Department='Electrical' WHERE Name='Jack';
Query OK, 1 row affected (0.12 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT * FROM tbl1 WHERE Name='Jack';
+----+------+------------+----------+
| ID | Name | Department | Password |
+----+------+------------+----------+
|  2 | Jack | Electrical | pass2    |
+----+------+------------+----------+
1 row in set (0.00 sec)
```

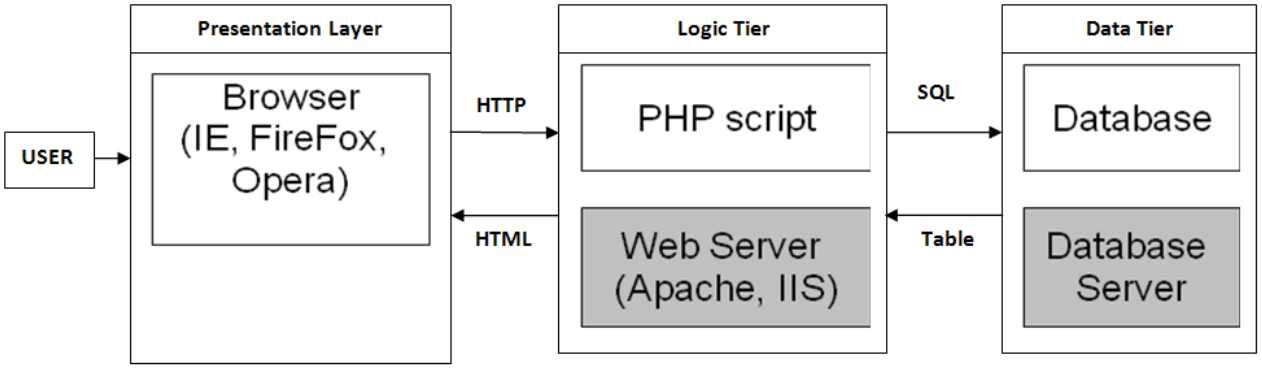
3. WEB APPLICATION ARCHITECTURE

Web application consists of three major components they are

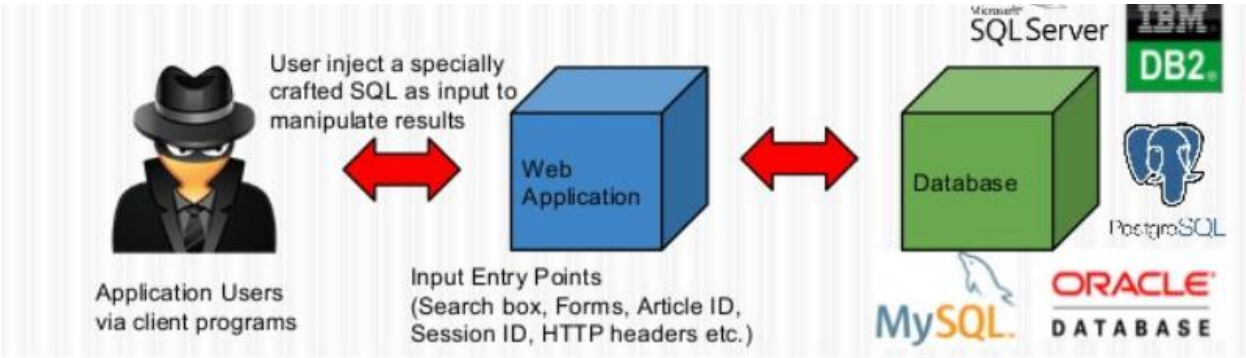
1.Web Browser – Client side – Communicates with user.

2.Web application and server – generates and delivers content to browser.

3.Database – stores data.



SQL Injection attacks causes severe damage to the database. Users communicate to the database with the help of Web servers. If these servers are not implemented properly there is a chance of SQL Injection attack.



3.1. Start Apache Server

It is a web server that hosts HTTP based websites. It supports multiple programming languages, server-side scripting, authentication mechanism and database support.

```
[12/05/18]seed@VM:~$ sudo service apache2 start
[sudo] password for seed:
[12/05/18]seed@VM:~$
```

4. LAB TASKS

We have learnt how to create databases, tables and INSERT, SELECT UPDATE from the rows.

Now it's time to perform actual SQL Injection attack. For this lab we need three tools, (1) Firefox Web Browser, (2) Apache web server, (3) Web application. The pre-built VM image provided to you has already installed all these applications for you.

Web application

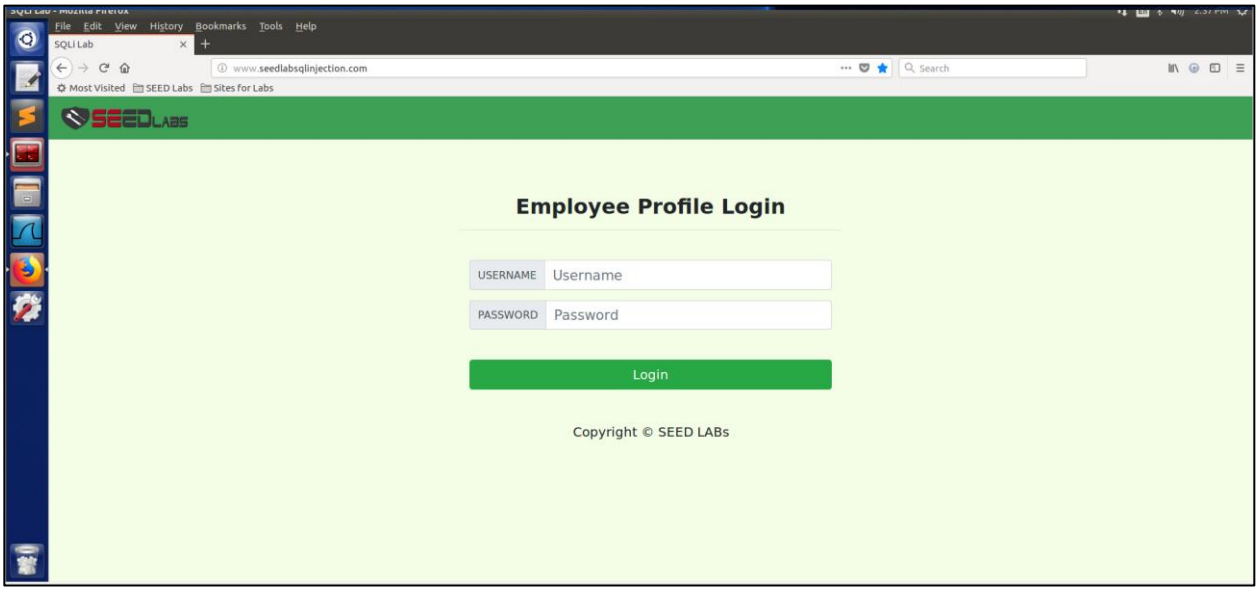
This VM has pre-built Employee Management web application for this lab. The web application is used to store employee profile information.

Employees can view and update their personal information in the database through this web application.

There are mainly two roles in this web application; Administrator is a privilege role and can manage each individual employees' profile information; Employee is a normal role and can view or update his/her own profile VM already has several employee accounts for this application.

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
1	Alice	10000	20000	9/20	10211002					fdbe918bdae83000aa54747fc95fe0470fff4976
2	Boby	20000	30000	4/20	10213352					b78ed97677c161c1c82c142906674ad15242b2d4
3	Ryan	30000	50000	4/10	98993524					a3c50276cb120637cca669eb38fb9928b017e9ef
4	Samy	40000	90000	1/11	32193525					995b8b8c183f349b3cab0ae7fccd39133508d2af
5	Ted	50000	110000	11/3	32111111					99343bff28a7bb51cb6f22cb20a618701a2c2f58
6	Admin	99999	400000	3/5	43254314					a5bdf35a1df4ea895905f6f6618e83951a6effc0

Open Firefox browser and go to t www.SEEDLabSQLInjection.com. You can see the below screen in your browser.



4.1. Task 1: Retrieve data from Application Database

The objective of this task is to get familiar with SQL commands by playing with the provided database. Database `Users` contains a table called `credential`; the table stores the personal information of every employee.

Please login to MySQL console using the following command

```
$ mysql -u root -pseedubuntu
```

load the Users database using following command

```
mysql> use Users;
```

use following command to print the tables in Users database

```
mysql> show tables;
```

```
[12/06/18]seed@VM:~$ mysql -u root -pseedubuntu
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use Users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------+
| Tables_in_Users |
+-----------------+
| credential      |
+-----------------+
1 row in set (0.00 sec)
```

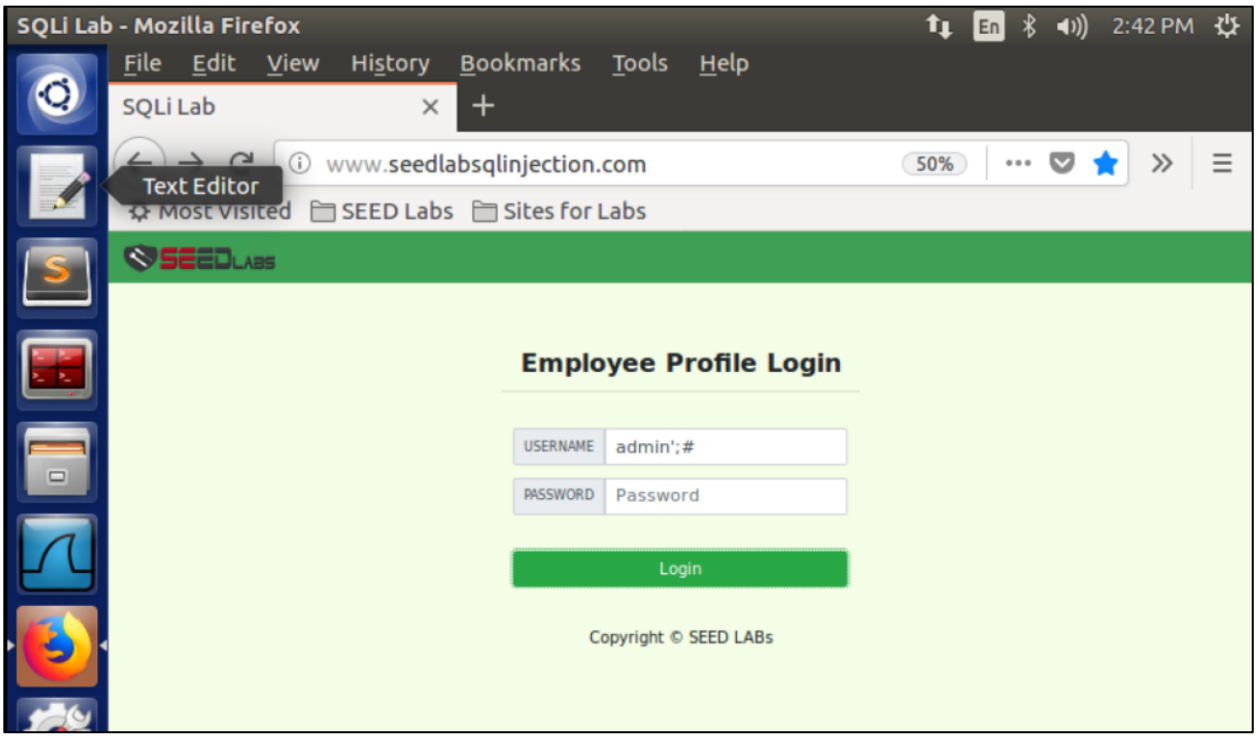
Now print all the profile information of employee Alice.

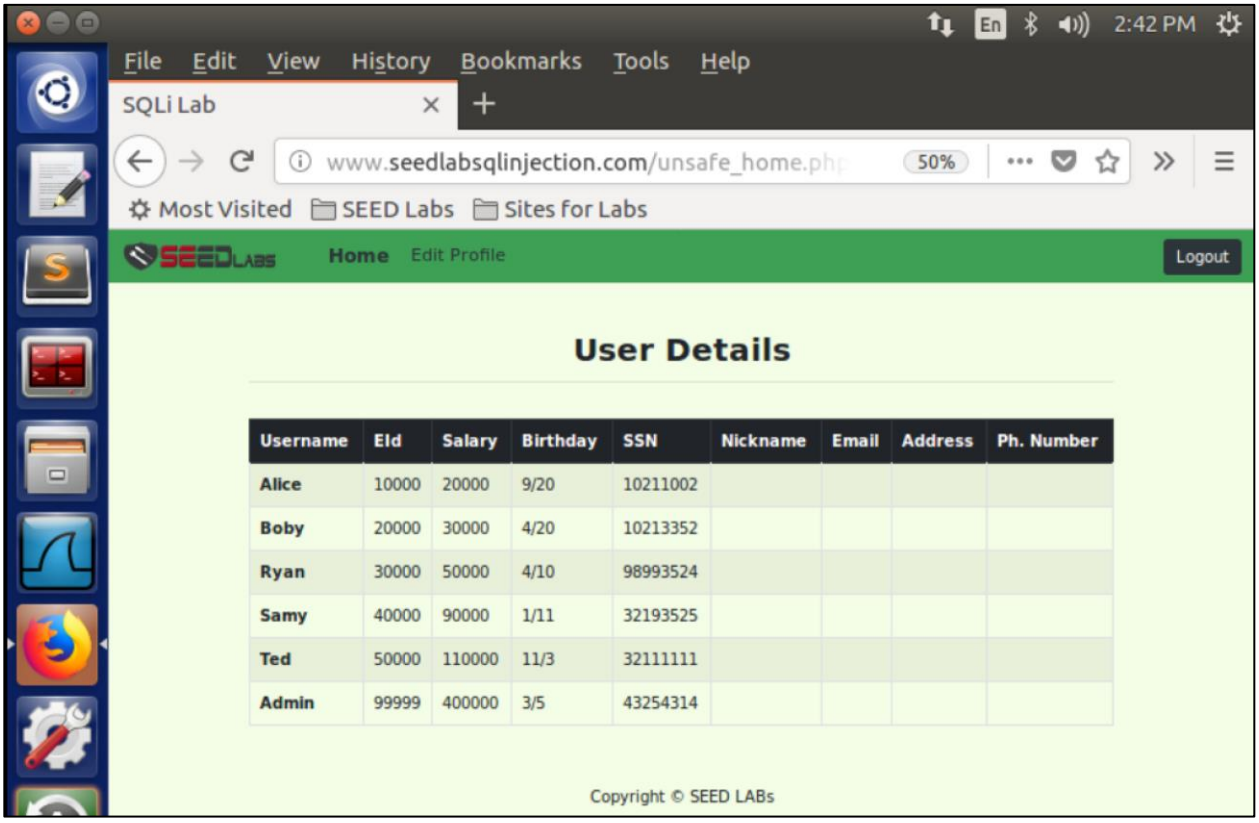
```
mysql> select * from credential where Name='Alice';
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
1 row in set (0.00 sec)
```

4.2. Task 2: SQL Injection Attack on SELECT Statement

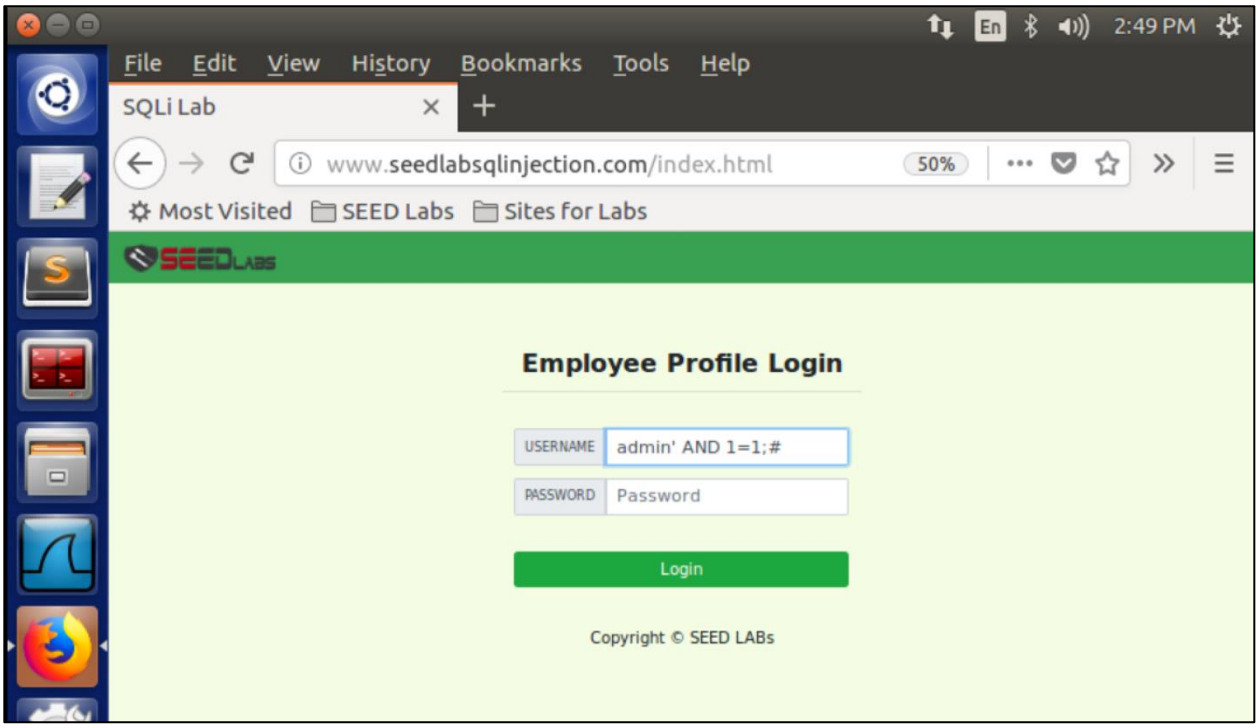
4.2.1: SQL Injection Attack from webpage

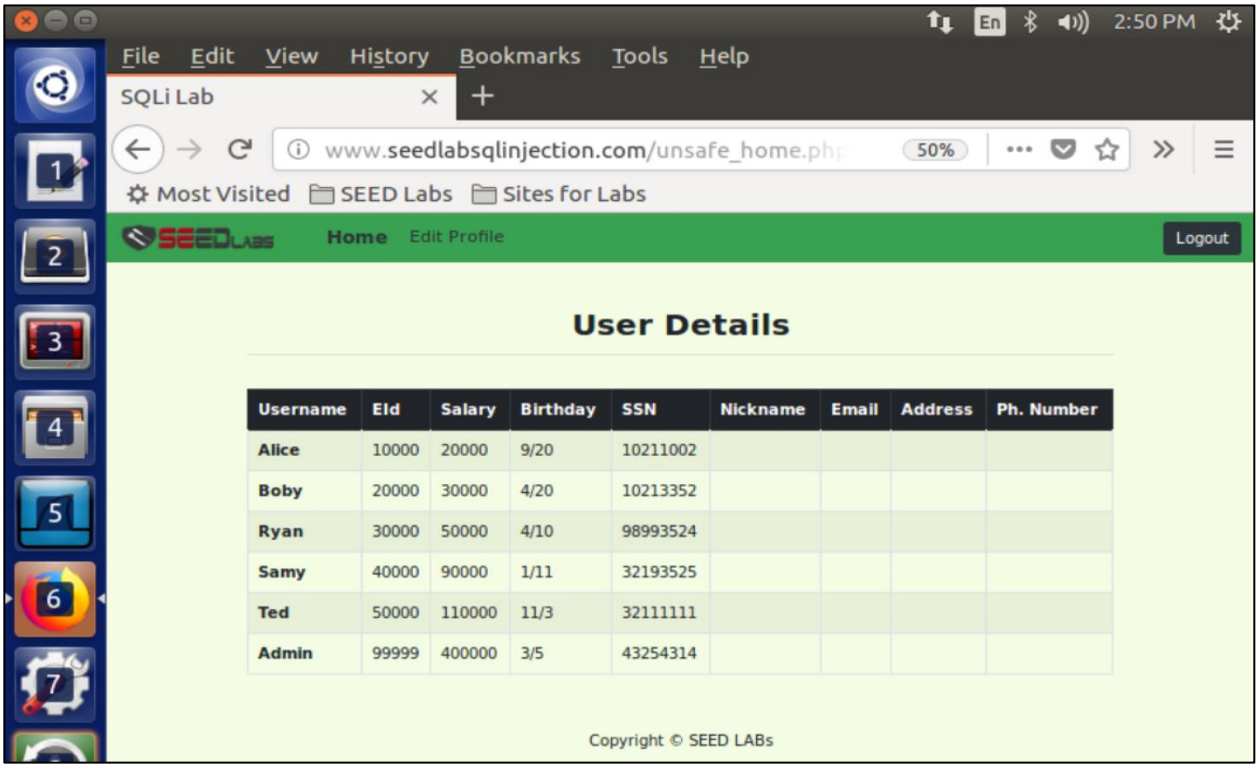
Our task is to login into the web application as the administrator from the login page. We assume that you do know the administrator's account name which is `admin`, but you don't know the ID or the password. We can execute the attack in the following way.





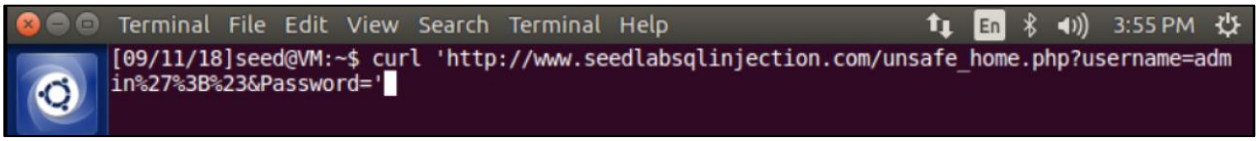
This attack can be done in other way.

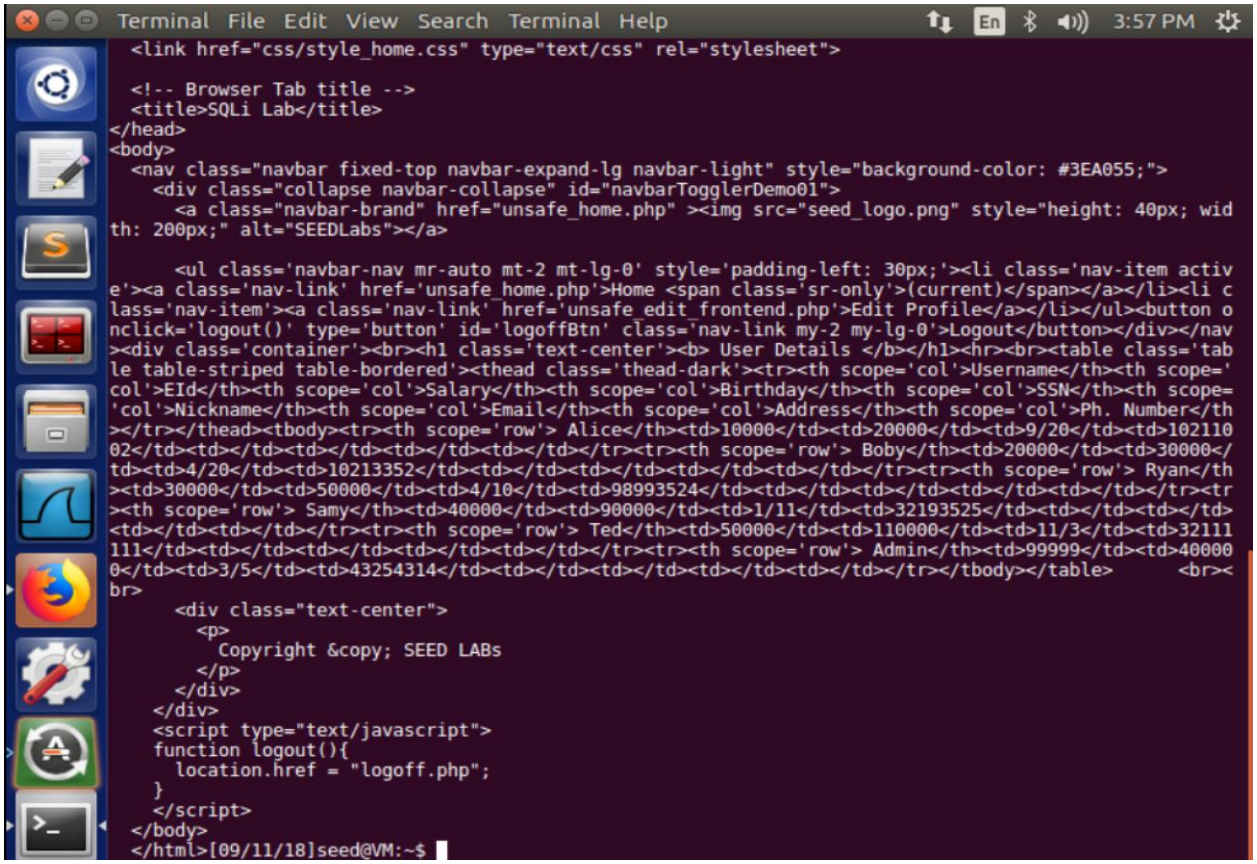




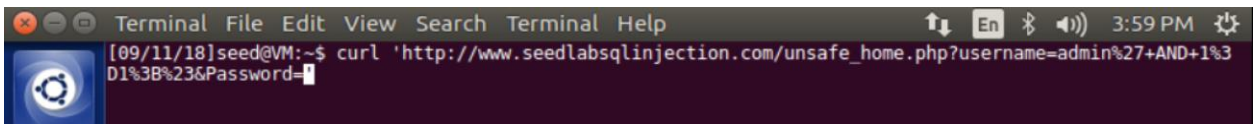
4.2.2: SQL Injection Attack from command line:

We are repeating the task 3.1 but here we are using command line tool such as `curl` to perform our task. `Curl` is a tool which can send HTTP requests, if you want to include multiple parameters in HTTP requests, you need to put URL and the parameters between a pair of single quotes; otherwise, the special characters used to separate parameters (such as &) will be interpreted by the shell program, changing the meaning of the command.

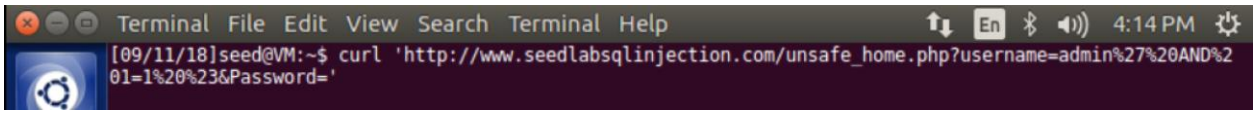




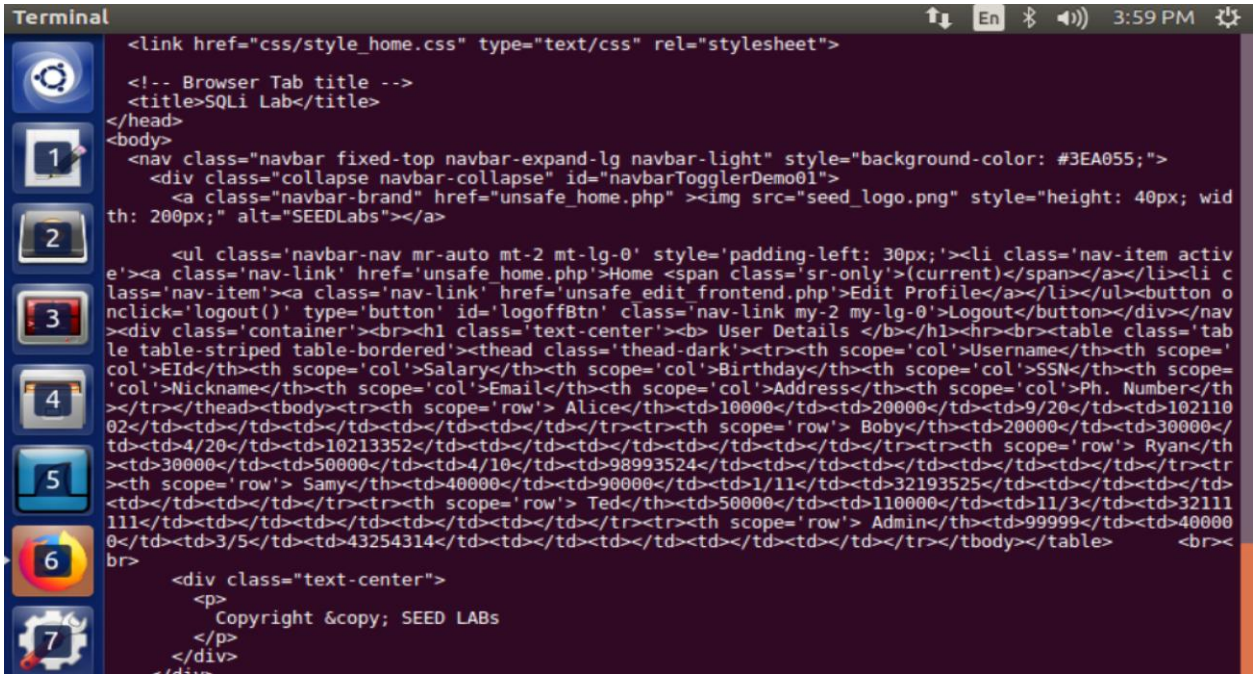
Another way it can be done:



OR

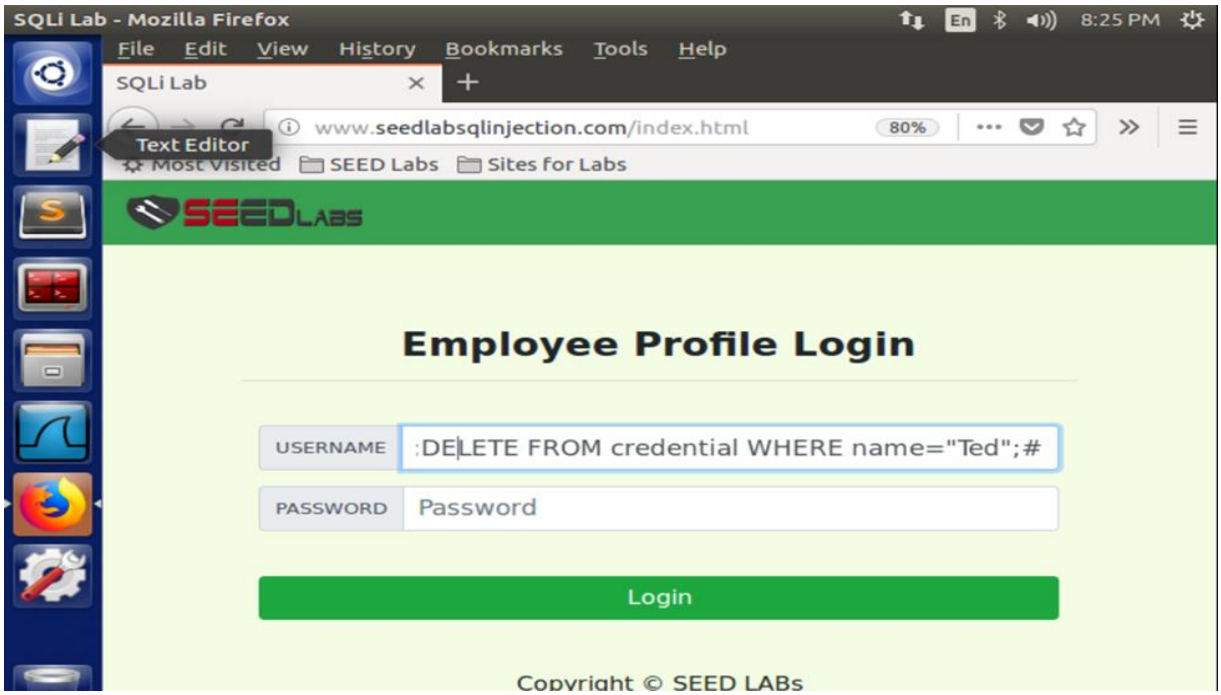


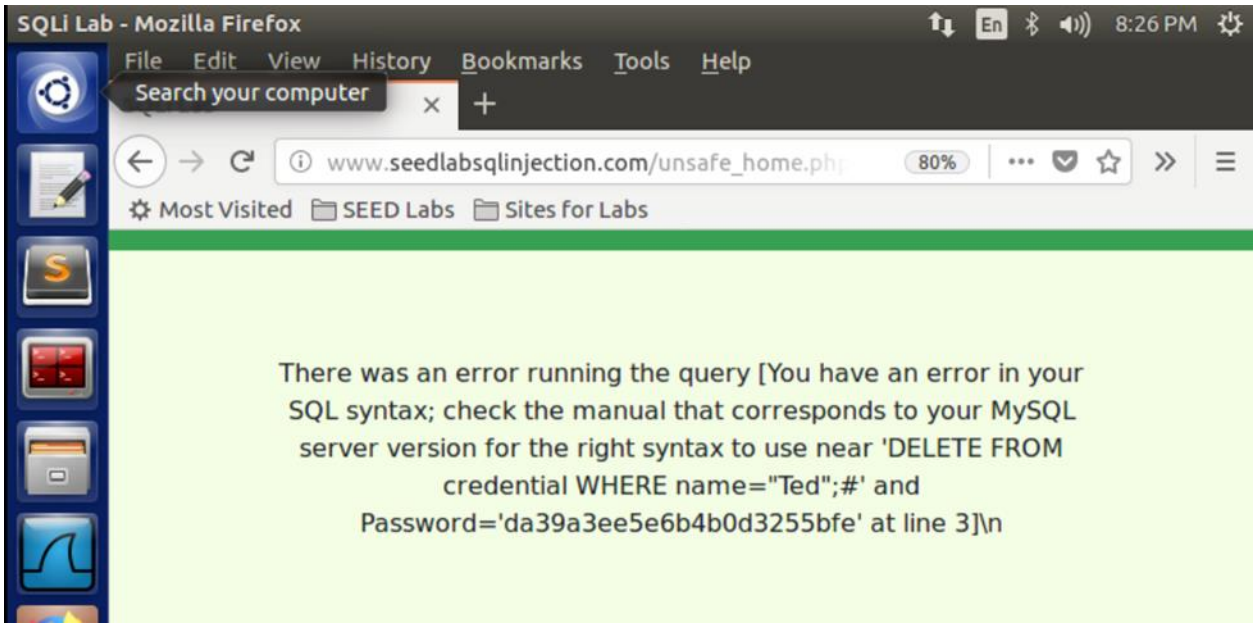
If you execute these command in the terminal, you should get the below output.



4.2.3: Append a new SQL Statement

In the above two attacks, we can only steal information from the database; it will be better if we can modify the database using the same vulnerability in the login page.



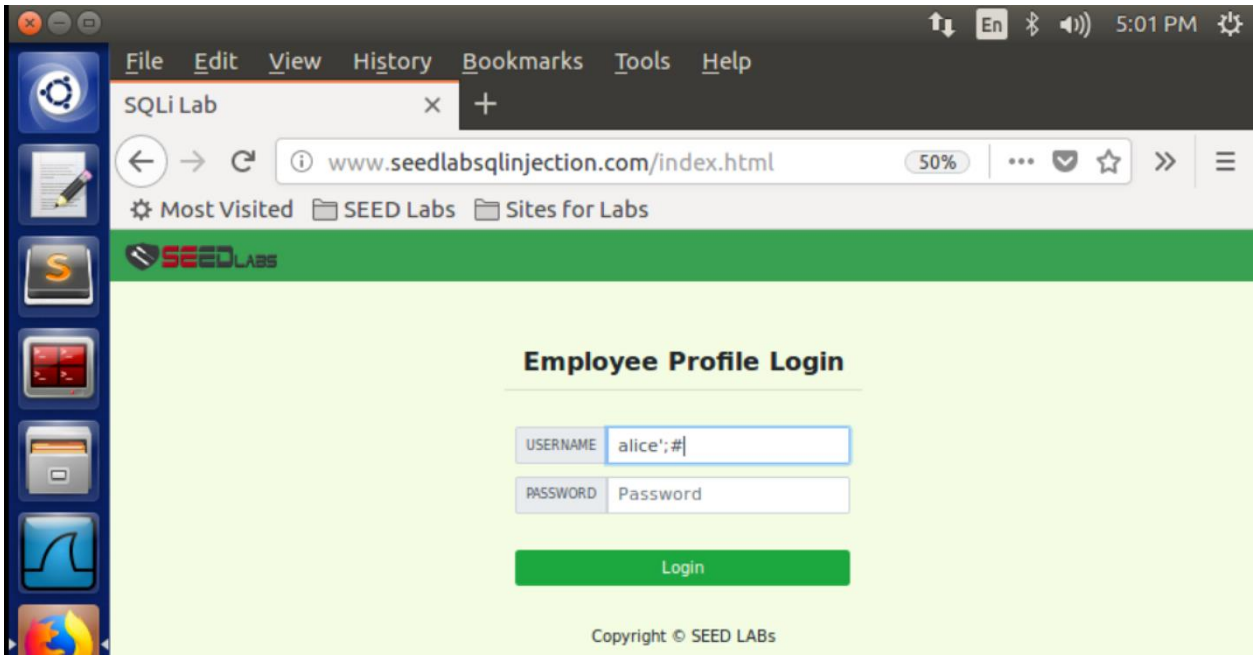


This is because the query() function used in the code accepts only one SQL statement. This is one of the protections against SQL Injection.

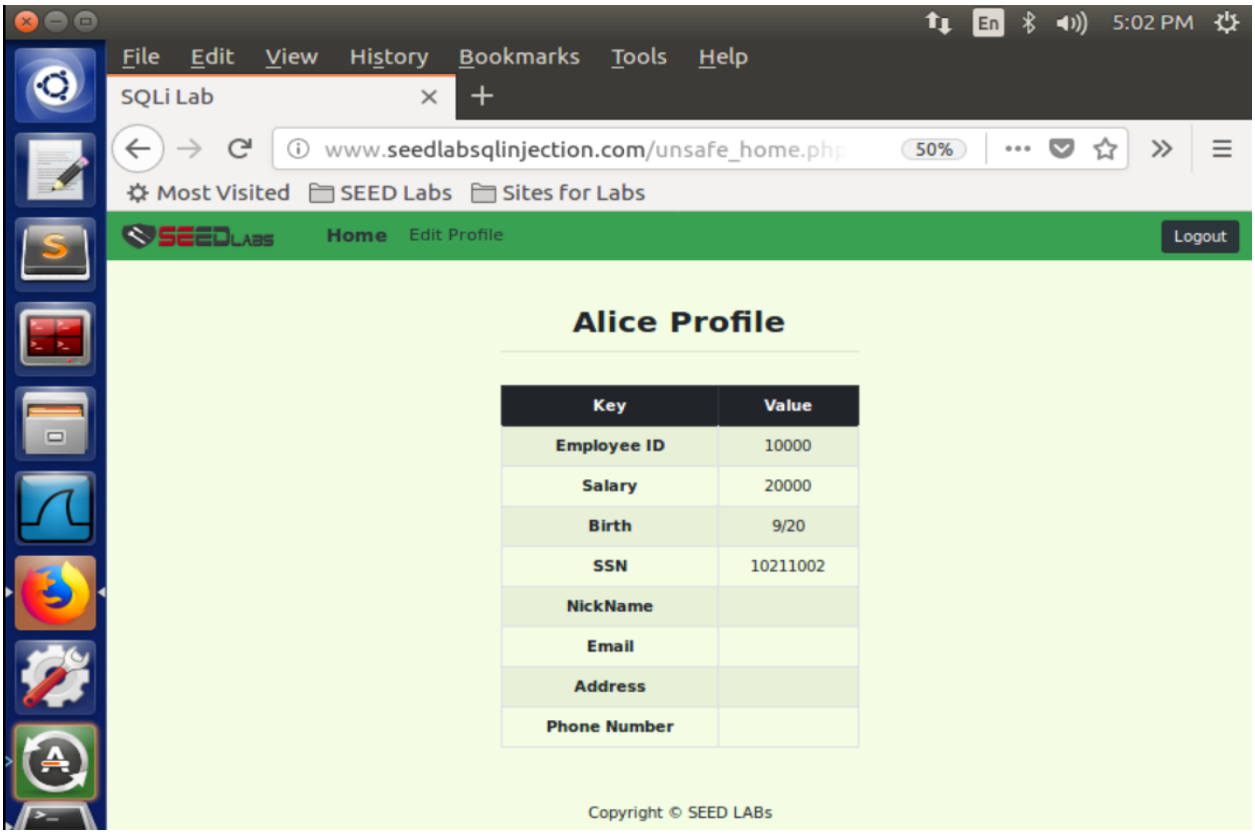
4.3. Task 3: SQL Injection Attack on UPDATE Statement

4.3.1. SQL Injection Attack on UPDATE Statement – modify salary:

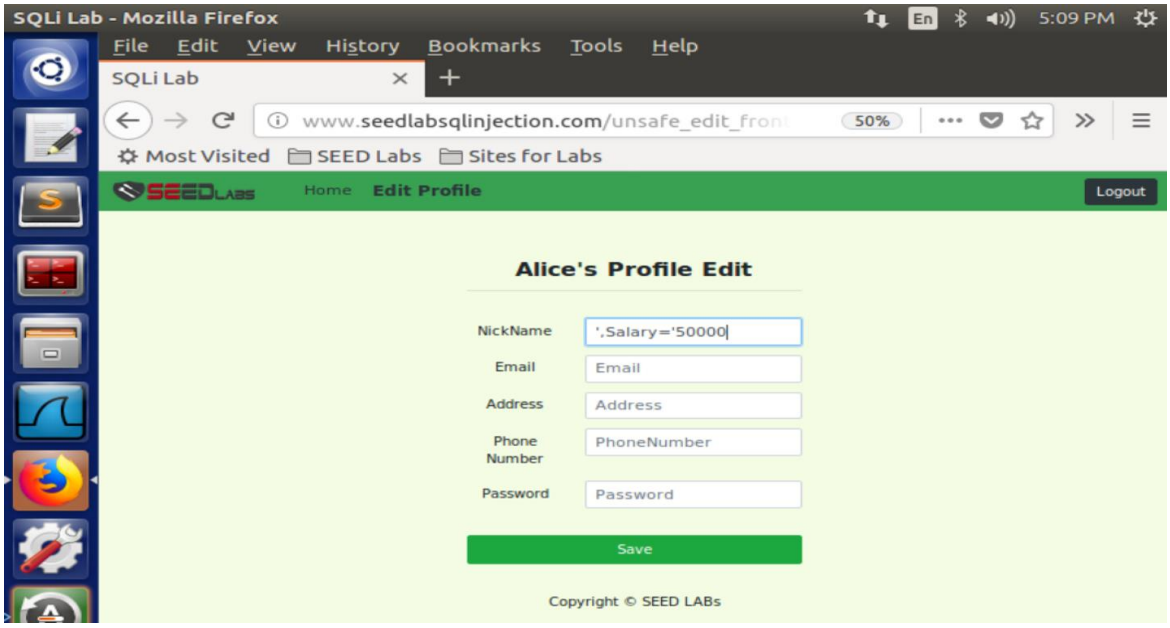
If a SQL injection vulnerability happens to an UPDATE statement, the damage will be more severe, because attackers can use the vulnerability to modify databases.

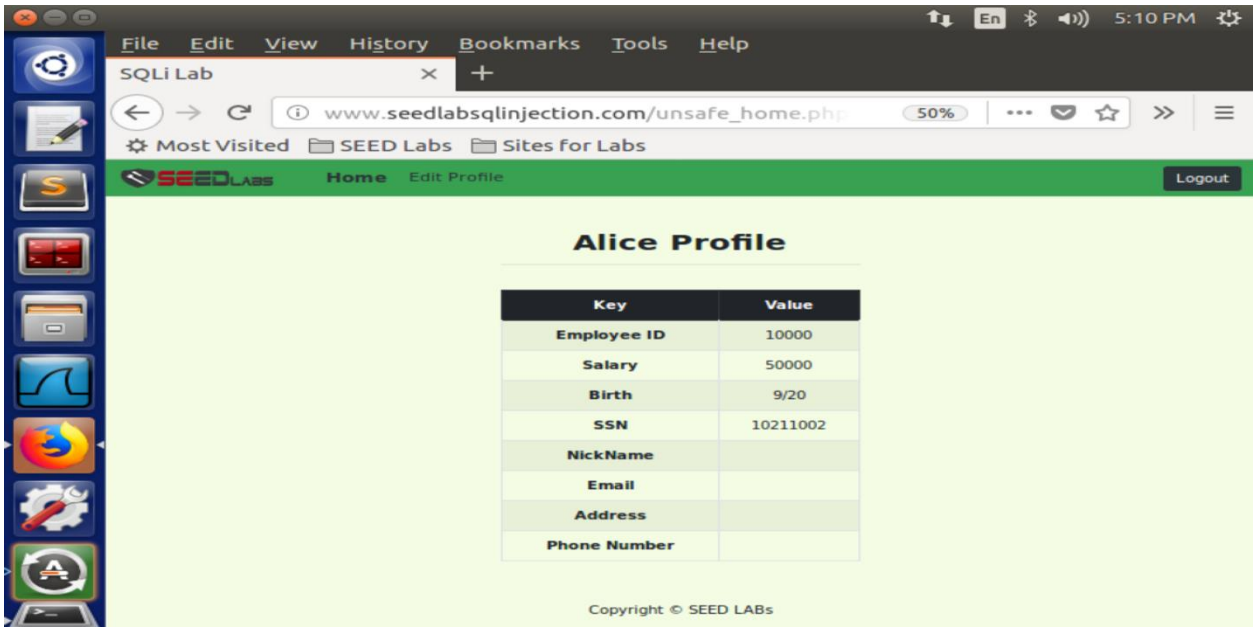


Now we are logged into Alice's profile. We can see her profile information like Employee ID, Salary and Date of Birth etc.,



In our Employee management application, there is an 'Edit Profile' page that allows employees to update their profile information, including nickname, email, address, phone number and password.

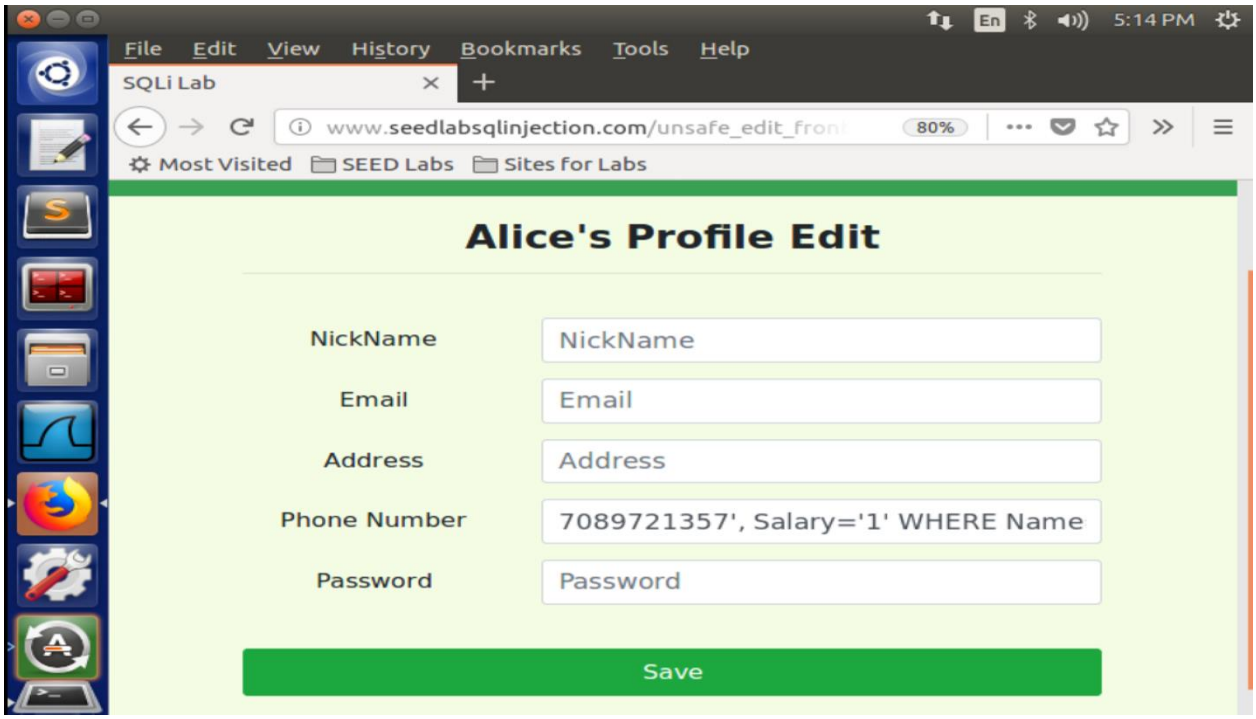


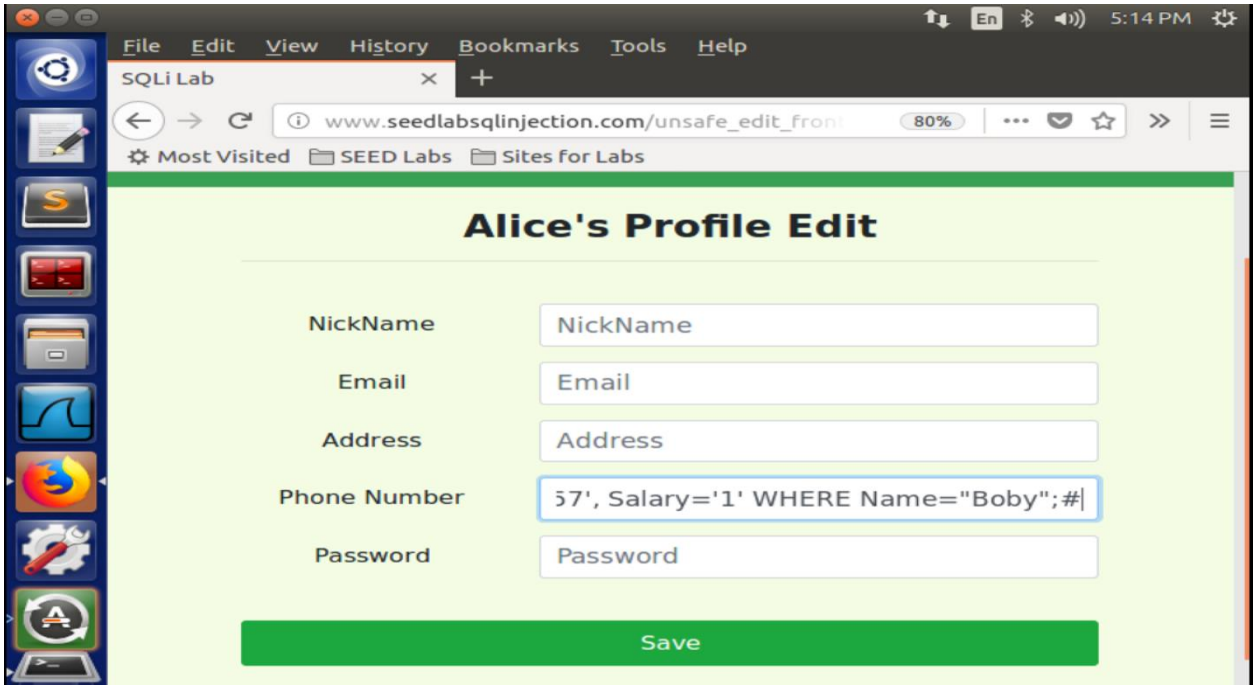


4.3.2. SQL Injection Attack on UPDATE Statement – modify other people' salary

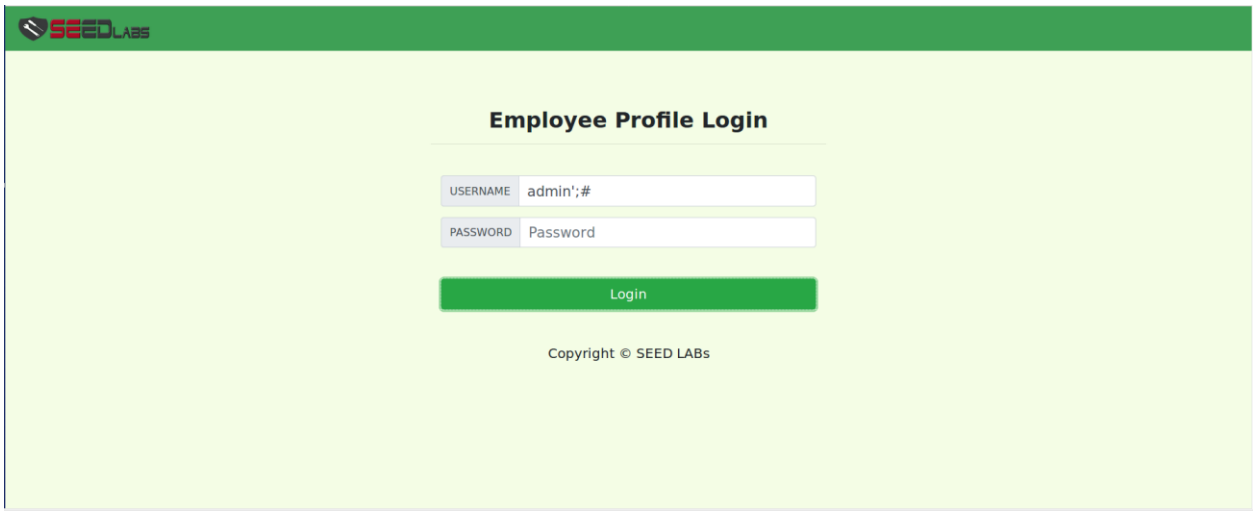
In this taks we are modifying Boby's salary to 1 dollar from Alice's edit profile page. We are using 'Phone number column' to enter a phone number and edit salary for Boby. Please enter the following query in the Phone Number field.

```
7089721357', Salary= '1' WHERE Name= "Boby";#
```

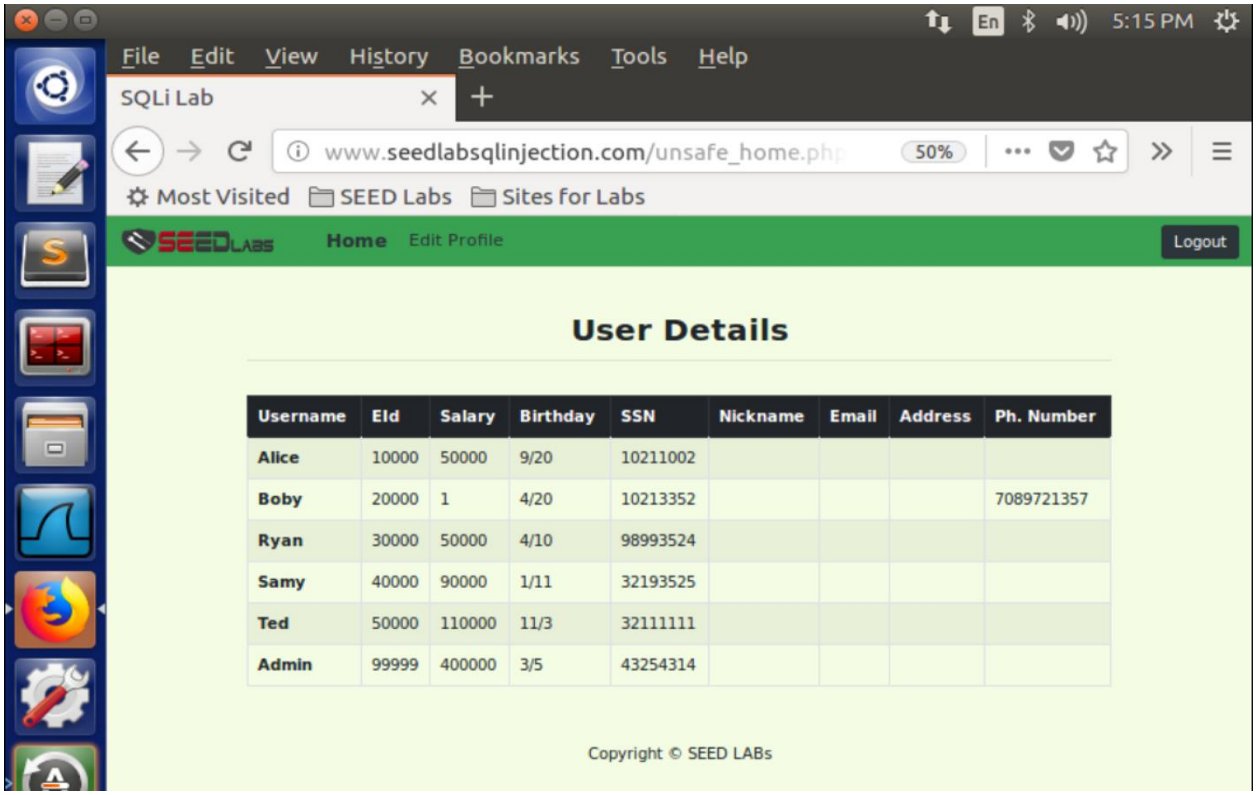




Now click on 'Logout' and you will be logged out from Alice's profile. You need to log in as administrator to see the changes made. For that go to 'Home' and log in as administrator as shown below.



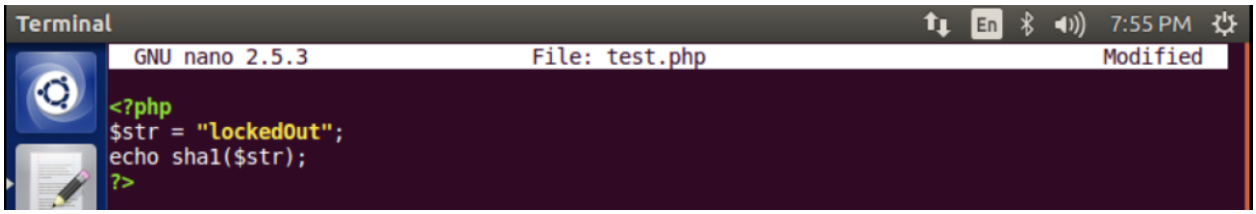
Here you can see the changes made to Boby's profile. Previously Boby's salary was 30000 now it is changed to one dollar.



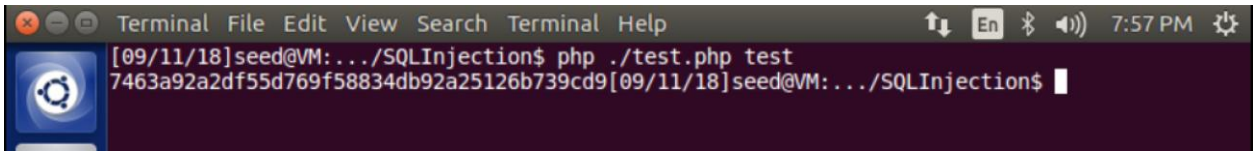
Username	Eid	Salary	Birthday	SSN	Nickname	Email	Address	Ph. Number
Alice	10000	50000	9/20	10211002				
Boby	20000	1	4/20	10213352				7089721357
Ryan	30000	50000	4/10	98993524				
Samy	40000	90000	1/11	32193525				
Ted	50000	110000	11/3	32111111				
Admin	99999	400000	3/5	43254314				

4.3.3. SQL Injection Attack on UPDATE Statement - Modify other people' password:

We created a php program to create sha1 hash of password ("lockedOut") to set for Boby. Go to nano editor and create a php file called test.php and pass the following code into the file.

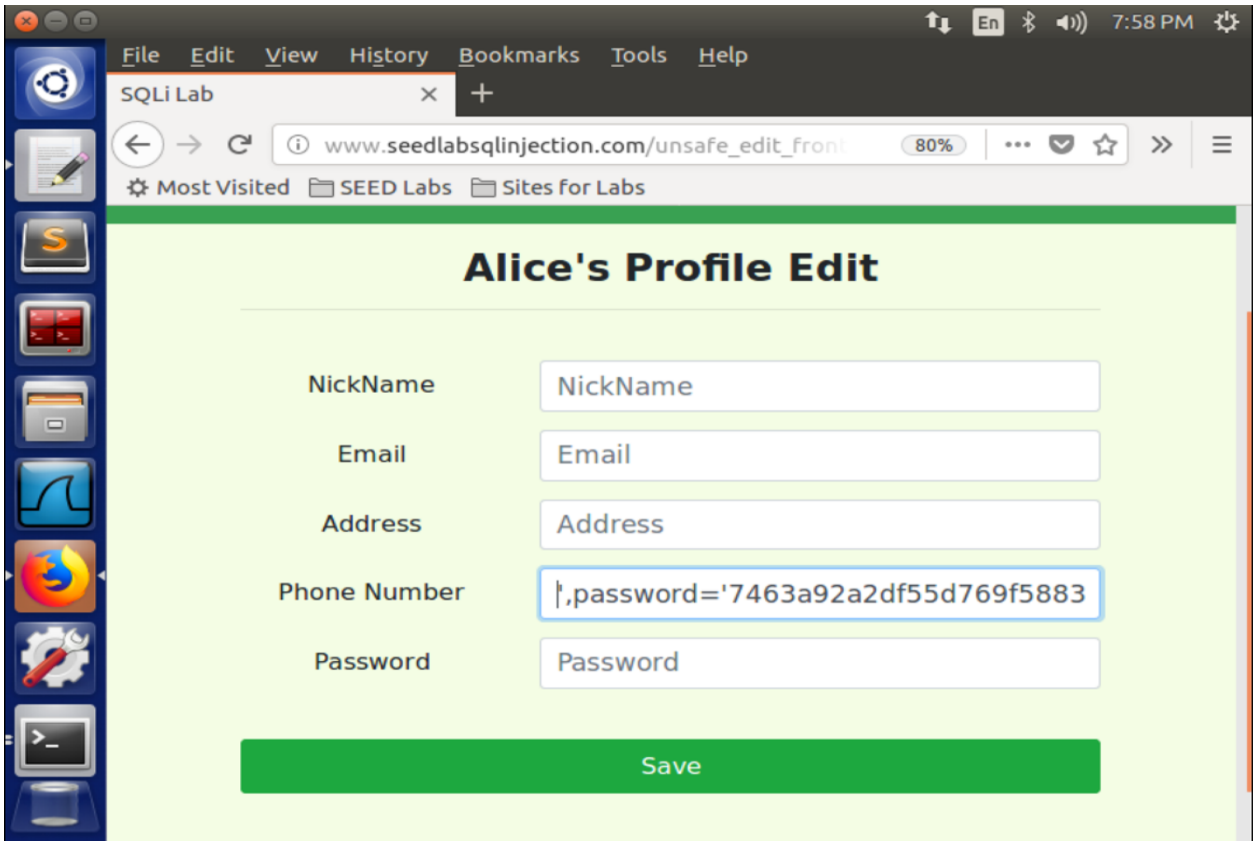


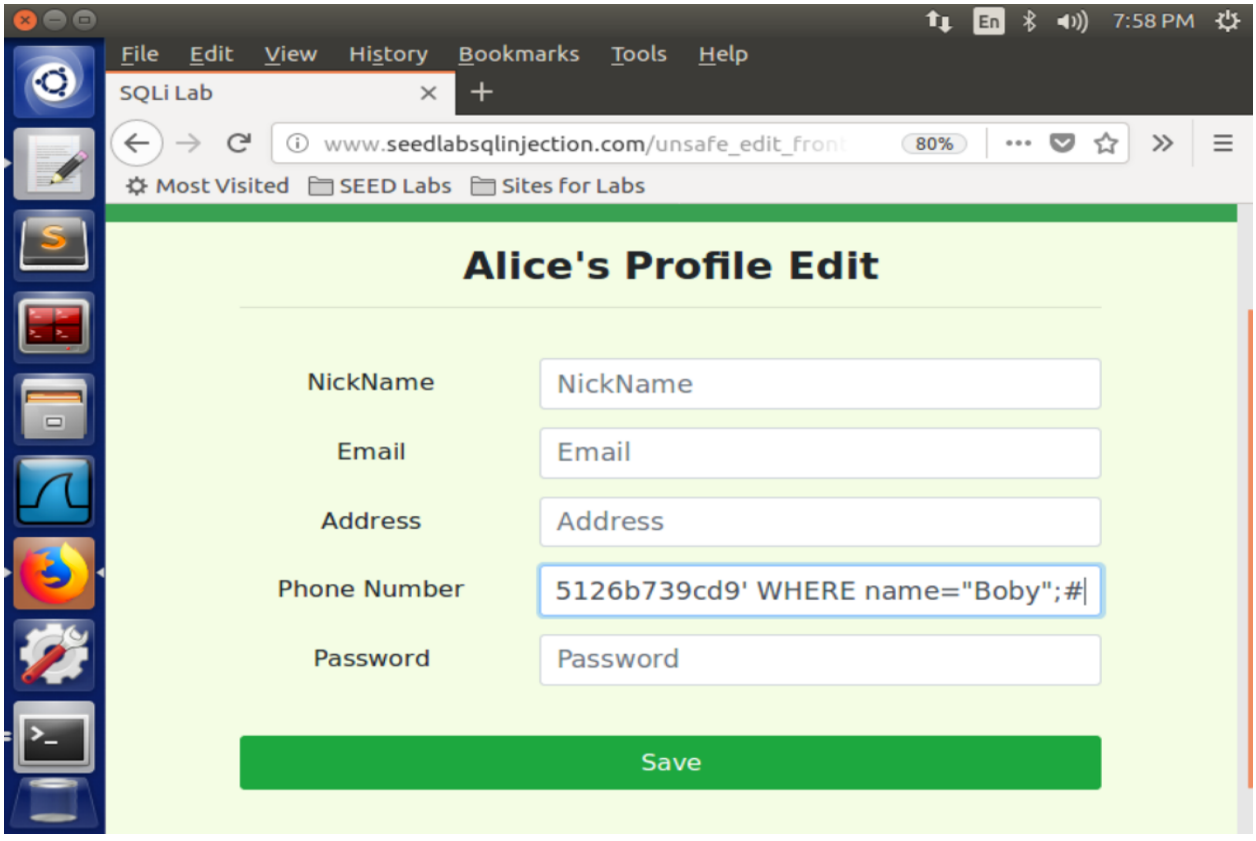
When we run the file, it produces Sha1 hash for "lockedOut":



Now we are creating the Injection to change the password for Boby through Alice's account. Go to Alice's edit profile section and enter the following statement in alice's phone number field.

```
',password='hashValue' WHERE name="Boby";#
```



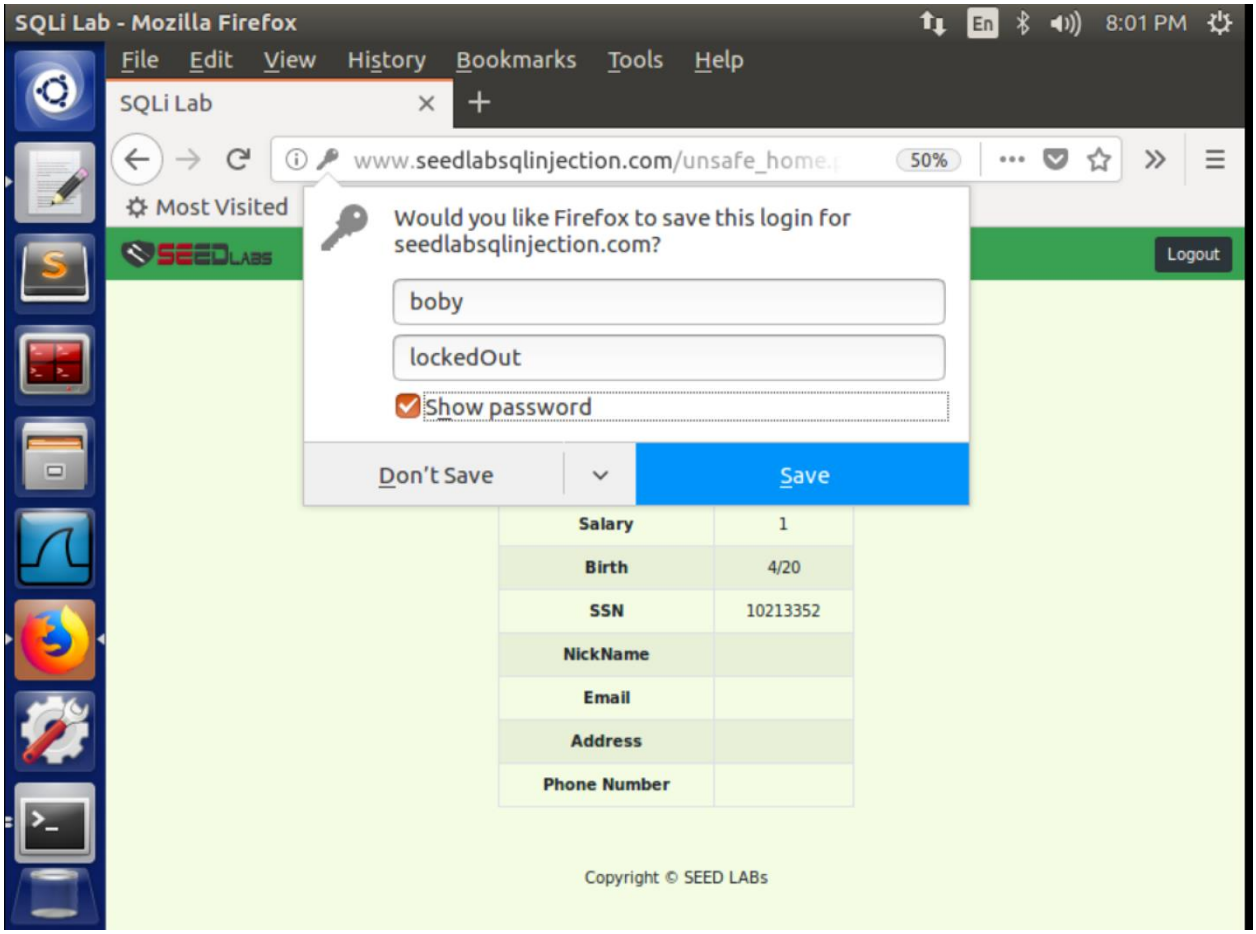


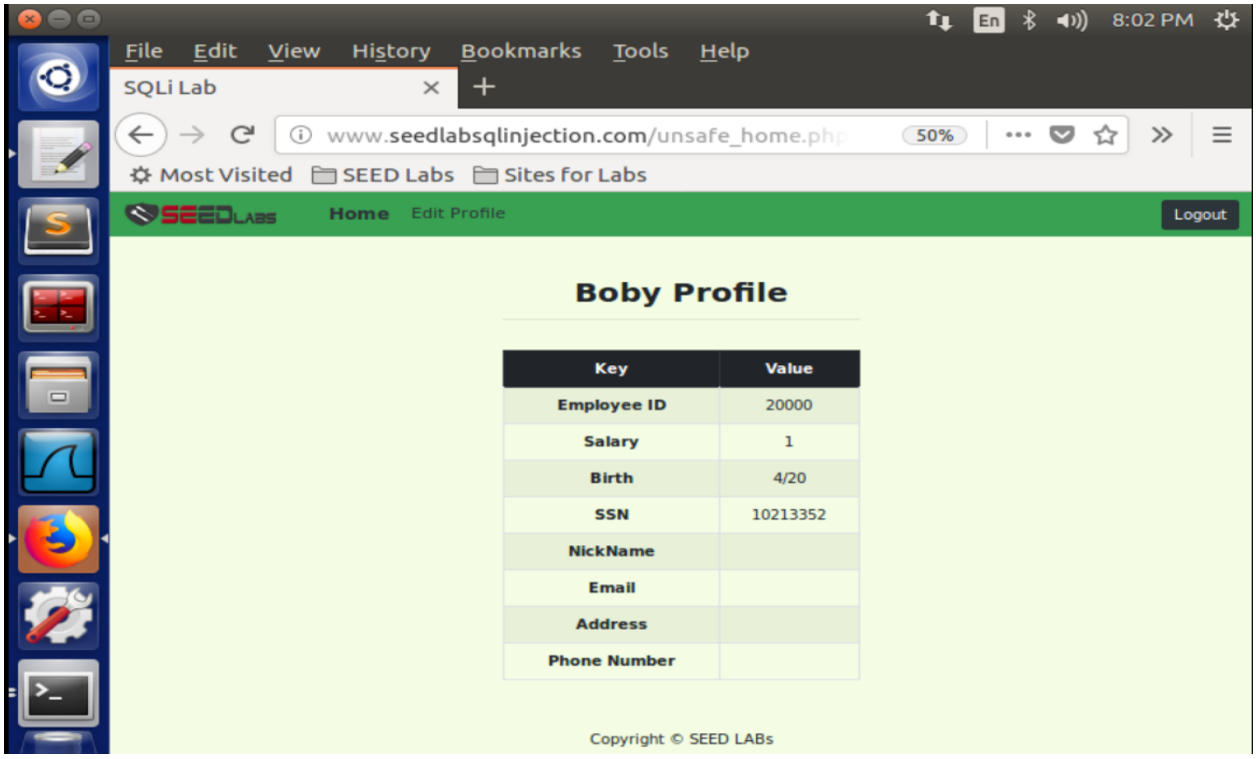
Then go to the terminal and open boby's record from the database using the following statement.

```
Mysql> SELECT * FROM credential WHERE Name="Boby";
```

Successfully changed the password and now, we can log in with username: "Boby" and Password: "lockedOut"

Go to browser and login with the new credentials.





WHAT TO SUBMIT

Please provide screen shots of the results obtained by running program examples. You are expected to make modifications to better understand how the attack can be generated and how the defense works.